



链滴

IPFS 数据链接与内容寻址

作者: [88250](#)

原文链接: <https://ld246.com/article/1557020262516>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

本文翻译自 IPFS 社区教程 [ProtoSchool](#)。ProtoSchool 是一个可以交互式学习 IPFS 编程的网站，及代码的部分大家可到该网站上直接运行测试！

本文将介绍如何在对等节点上创建和获取可验证的数据链接。

LESSION 1 - 创建节点并返回内容标识符 (CID)

在本教程中，我们将探索 IPFS DAG API，它允许我们将数据对象存储在 IPFS 中，你可以在 IPFS 中存储一些令人兴奋的东西，比如你最喜欢的小猫的 GIF。

你可以将数据对象作为参数传入 `ipfs.dag.put` 方法来创建新节点，该方法返回新创建的节点的内容标识符 (CID)。

```
ipfs.dag.put({ hello: 'world' })
```

CID 是 IPFS 中从其内容生成的数据地址。每当有人将相同的 `{hello: 'world'}` 数据放入 IPFS 时，他将获得与你获得的相同的 CID。如果他们改为输入 `{hello: 'wOrld'}`，则 CID 会有所不同。

如下示例代码将创建 `{ test: 1 }` 节点并返回其 CID：

```
const run = async () => {  
  let cid = await ipfs.dag.put({ test: 1 })  
  return cid  
}  
  
return run
```

LESSION 2 - 链接到已有节点

有向无环图 (DAG) 的一个重要特性就是可以相互链接。链接的方式就是存储相关节点的 CID。

例如一个节点存有一个 `foo` 指向另一个 CID 节点 `barCid`：

```
{  
  foo: barCid  
}
```

上述示例我们创建了一个 `foo` 字段并将其值链接到了另一个 CID，我们将其称之为命名链接。

我们可以将命名链接保存到 IPFS 中：

```
await ipfs.dag.put({ foo: barCid })
```

如下示例代码将创建 `{ test: 1 }` 并将其命名为 `bar` 链接到新创建的节点上：

```
const run = async () => {  
  let cid = await ipfs.dag.put({ test: 1 })  
  let cid2 = await ipfs.dag.put({ bar: cid })  
  return cid2  
}  
  
return run
```

LESSION 3 - 通过链接读取嵌套数据

我们可通过路径查询读取嵌套数据。

```
let cid = await ipfs.dag.put({
  my: {
    deep: {
      obj: 'is cool'
    }
  }
})
```

```
console.log(await ipfs.dag.get(cid, '/my/deep/obj'))
// prints { value: 'is cool', remainderPath: '' }
```

`ipfs.dag.get` 可 IPFS 路径进行查询，查询结果包含了结果值以及尚未解析的路径。我们可以通过该 API 进行遍历：

```
let cid = await ipfs.dag.put({ foo: 'bar' })
let cid2 = await ipfs.dag.put({
  my: {
    other: cid
  }
})
```

```
console.log(await ipfs.dag.get(cid2, '/my/other/foo'))
// prints { value: 'bar', remainderPath: '' }
```

```
// Notice above how this method returns not the value itself,
// but an object that contains a `value` property
```

如下示例代码将返回 `test` 的值：

```
const run = async () => {
  let cid = await ipfs.dag.put({ test: 1 })
  let cid2 = await ipfs.dag.put({ bar: cid })
  let cid3 = await ipfs.dag.get(cid2, '/bar/test')
  return cid3.value
}

return run
```